NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

AN ATTACKER-DEFENDER MODEL FOR IP-BASED NETWORKS

by

Timothy R. Barkley

March 2008

Thesis Advisor:	David L. Alderson
Second Reader:	W. Matthew Carlyle

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
March 2008

3. REPORT TYPE AND DATES COVERED
Master's Thesis

4. TITLE AND SUBTITLE An Attacker-Defender Model for IP-Based Networks

5. FUNDING NUMBERS

6. AUTHOR(S) Barkley, Timothy Ruben, U. S. Navy

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE
A

13. ABSTRACT (maximum 200 words)

The Internet Protocol (IP) has emerged as the dominant technology for determining how data is routed across the Internet. Because IP flows are defined essentially in terms of origin-destination (O-D) pairs, we represent IP traffic engineering as a multi-commodity flow problem in which each O-D pair is treated as a separate commodity. We account for the diversity in IP routing by modeling opposite extremes of traffic engineering: "naive" traffic engineering where the IP routes data between any two users using only the shortest path between them, and "best case" traffic engineering where IP has the flexibility to route data using multiple paths in the network regardless of their length. We develop linear programming formulations that identify the maximum data flow for an IP network that satisfies proportionality constraints for traffic demand for each case of traffic engineering, and we also determine the optimal interdiction of those flows that reduces that maximum flow in the worst possible way.

14. SUBJECT TERMS Internet Service Provider, Internet Protocol, Traffic Engineering, Linear Programming, Networks, Multi-commodity Max Flow, Interdiction

15. NUMBER OF PAGES
79

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

THIS PAGE INTENTIONALLY LEFT BLANK

AN ATTACKER-DEFENDER MODEL FOR IP-BASED NETWORKS

Timothy R. Barkley
Lieutenant, United States Navy
B.S., Virginia Tech, 2001

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS ANALYSIS

from the

NAVAL POSTGRADUATE SCHOOL
March 2008

Author: Timothy R. Barkley

Approved by: David L. Alderson
Thesis Advisor

W. Matthew Carlyle
Second Reader

James Eagle
Chair, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The Internet Protocol (IP) has emerged as the dominant technology for determining how data is routed across the Internet. Because IP flows are defined essentially in terms of origin-destination (O-D) pairs, we represent IP traffic engineering as a multi-commodity flow problem in which each O-D pair is treated as a separate commodity. We account for the diversity in IP routing by modeling opposite extremes of traffic engineering: “naive” traffic engineering where the IP routes data between any two users using only the shortest path between them, and “best case” traffic engineering where IP has the flexibility to route data using multiple paths in the network regardless of their length. We develop linear programming formulations that identify the maximum data flow for an IP network that satisfies proportionality constraints for traffic demand for each case of traffic engineering, and we also determine the optimal interdiction of those flows that reduces that maximum flow in the worst possible way.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

The objective of this thesis is to provide a quantitative means to assess the carrying capacity of an Internet Protocol (IP) based network under a general model for traffic demands, as well as identify the node and/or arc attacks that interrupt traffic flows in the worst possible manner.

Over the last decade the Internet has become a critical infrastructure to our way of life. Internet Service Providers are the owners and operators of the computer networks that collectively afford the general public, schools, businesses, government, and military organizations, access to the Internet and its evolving applications. Network operators have developed explicit and implicit mechanisms for influencing the way in which IP traffic travels across their networks. This process is known as traffic engineering.

We formulate a model representing “naive” traffic engineering where IP routes data for each origin-destination pair using only a single shortest path in the network. We desire to maximize this total amount of data flow by raising flow along every path in a proportional manner until one of the internal nodes and/or connecting arcs reaches capacity. Next we formulate a model representing “best case” traffic engineering where IP has the flexibility to route data using multiple paths in the network regardless of length. We maximize the sum of the flows on artificial return arcs by increasing flow along all of them in proportion to each other until one of the arcs in the network reaches its capacity.

ISPs are susceptible to many types of attacks, both physical and cyber, to their key components. The models developed here identity locations of attacks that have the most negative impact on the performance of the ISP.

The analysis here focuses on Abilene, the high-speed backbone of the Internet2 educational network, a not-for-profit advanced networking consortium of universities, laboratories, and government agencies. We perform our analysis on a network representation of Abilene with node and/or arc capacities. We compute

the total amount of traffic routed between customers, the overall flow through the network, and the utilization of Abilene's transshipment routers using both the naive and the best case traffic engineering formulations. We also identify the optimal node and arc attacks that affect the total amount of traffic routed between customers and the flow through the network in the worst possible way. We find that Abilene is well-provisioned in the sense that it tends to be the arcs, in particular the customer access links, that saturates data flow in the network, a generalization that is consistent with our results.

The models and analysis in this thesis are applicable to any ISP network. The general public, businesses, civilian and military organizations rely heavily on these networks. As the reliance grows, so will the need for understanding an ISP's limitations and vulnerability to attacks.

ACKNOWLEDGMENTS

During the last two years, I have been fortunate to obtain a Master's in Science in Operations Analysis from the Naval Postgraduate School. I feel that the education I received here at NPS is rewarding, and provides me with a strong foundation for when I return to the fleet, as well as my career after the Navy.

As I prepared my thesis, there were several people that were instrumental in helping me through this demanding and challenging process. I appreciate the assistance of NPS professors Matthew Boensel, Matthew Carlyle and Gerald Brown. I would especially like to thank my advisor David Alderson for his knowledge, guidance, and confidence in me. I am extremely proud of the work that we have done! And last but certainly not least, I want to thank my family, especially my mom Priscilla Barkley and my brothers Steve and Mario Barkley, for their continued love and support.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

Over the last decade the Internet has become an infrastructure that is critical to supporting our way of life. People throughout the world rely on the Internet as a means for personal communication through the use of email, instant messaging, or chat rooms. Students have access to limitless amounts of information stored on the Internet on any topic imaginable. Co-workers are able to share information and conduct business in unprecedented manners. The Navy Marine Corps Intranet and the Army's LandWarNet provide service members on all command levels with secure platforms for information sharing amongst military installations and forward-deployed forces throughout the world.

Internet Service Providers (ISPs) are the owners and operators of the computer networks that collectively provide the general public, schools, businesses, government, and military organizations, access to the Internet and its evolving applications.

The operation of the Internet is determined by protocols which specify the roles, rules, and responsibilities for individual technologies. Among these, the Internet Protocol (IP) has emerged as the dominant technology for determining how an ISP routes traffic across its part of the Internet from one customer to another.

Network operators have developed explicit and implicit mechanisms for influencing the way in which IP traffic travels across their networks. This process, known as "traffic engineering," allows the network operator to tune the performance of their network in response to changing traffic levels or environmental conditions. The two main protocols used to for traffic engineering of IP within a single ISP are Open Shortest Path First (OSPF) and Intermediate

System-Intermediate System (IS-IS), both of which compute shortest paths based on configurable link weights (see Rexford, 2006 and references therein).

B. RESEARCH OBJECTIVES AND MODELING APPROACH

The objective of this thesis is to provide a quantitative means to assess the carrying capacity of an IP-based network under general traffic demands, and then to identify the node and/or arc attacks that interrupt traffic flows in the worst possible manner. Such tools will lead to a better understanding of the system-wide vulnerabilities of real IP networks, as well as provide guidelines for network protection. We measure the performance of a given network in terms of the maximum traffic levels that it can support. We identify network vulnerabilities by determining the attack(s) to network components that reduce its maximum carrying capacity in the worst possible way.

We represent IP traffic flow using a "gravity model" for traffic demand, which states that the amount of traffic exchanged between two users is proportional to the total amount of traffic entering and exiting each of those users (Alderson et al., 2006). Thus, the gravity model assumes that demand for traffic is proportional to the product of the "size" of the two users. In practice, the actual traffic levels (i.e., data flow between users) need not be proportional, even when the demands follow the gravity model. However, we assume that traffic levels occur in proportion to demand, which is an extreme type of "fairness" that we impose. The idea is to provide a share of network resources (e.g., transshipment router bandwidth throughput capacity) to each user based on their size.

IP traffic engineering varies from ISP to ISP and depends on the technologies and polices in use. For example, it may be the intent of the ISP to minimize end-to-end traffic delay, or maximize utilization of network resources, or maximize "customer satisfaction." Some ISP users may receive preferential access to network resources, with the other users sharing what remains. So we model two opposite extremes of traffic engineering alternatives. We first formulate a model representing "naive" traffic engineering where IP routes data for each origin-destination pair using only a single shortest path in the network. This policy is easy to implement but tends to underutilize network resources. The second formulation represents “best case” traffic engineering where IP has the flexibility to route data using multiple paths in the network regardless of length. This policy yields a higher utilization of resources but is more complicated to implement and manage, and is an upper bound on achievable performance.

We represent a particular ISP as a network by considering its router-level map. Nodes in the network correspond to routing devices, and arcs between routers correspond to direct connectivity as seen by IP. For simplicity, we assume that connections between nodes correspond to physical connectivity, although this may not be the case. We also consider the network capacities in the form of connection speeds for arcs, and router throughput bandwidth capacities (Alderson et al., 2005).

We develop linear programming (LP) models that allow us to analyze the maximum carrying capacity of an ISP under a gravity model of user traffic demand. The models also examine the utilization of the ISP’s components (i.e., routers and their arcs), as well as identify the bandwidth limitations on those components. ISPs are susceptible to many types of attacks, both physical and cyber, to their key components (Doyle et al., 2005). The models developed here identify the attacks that have the biggest impact on the performance of the ISP.

C. LITERATURE REVIEW OF PREVIOUS WORK

The study of network vulnerability problems is not new. For telecommunications, considerable effort has been directed at the analysis of the physical infrastructure, in particular the design of fiber optic networks (Henningsson et al., 2006). Grotschel et al. (1995) present a general framework for the design of “survivable” communication networks, including the study of minimum spanning trees, Steiner trees, and minimum cost *k*-connected network design problems. An updated treatment of the problem can be found in Kerivin and Mahjoub (2005).

Much of the work in network vulnerability and survivability has its roots in graph theory, in which the network is represented solely in terms of its connections (without any annotations or domain-specific data), and considerable effort is devoted to assessing various measures of global connectivity. These include network diameter (i.e., average length of shortest path between any two nodes), characteristic path length (i.e., the average distance along any path between any two nodes), or the size and distribution of connected clusters. Recently, these graph theoretic measures have been applied to the Internet, and many studies have focused on how these connectivity patterns change in the presence of accidental or intentional graph losses (Albert et al., 2000, Cohen et al., 2000, Cohen at al., 2001, Bollobas and Riordan 2003, Crucitti et al., 2004). As discussed in Alderson (2008), a general problem with this approach is that any notion of network "function" is being approximated (often poorly) by these simple graph theoretic measures.

The vulnerability of router-level Internet networks was discussed by Doyle et al. (2005), who showed that previous results by Albert et al. (2000), which focused on connectivity patterns and focused on critical high-degree hubs, were not relevant to the real Internet. In contrast, they considered the need to maximize flow on the part of the ISP and formulated a simple path-based model of network throughput, described here as the "single-path" model. However, their consideration of "worst case" attacks on network routers was myopic and heuristic, in that it simply ranked nodes in a prioritized list in terms of the effect their removal would have on overall network throughput. They did not consider attacks that were formally optimal, nor did they consider more sophisticated models of traffic engineering that underlie real IP networks.

Recent effort has been devoted to the application of optimal network interdiction to critical infrastructure protection (Brown at al., 2006). This thesis continues that effort and formalizes the notion of an optimal attack for a maximum proportional flow problem and provides analysis and computational implementation to solve it efficiently.

D. STRUCTURE OF THESIS AND CHAPTER OUTLINE

The reminder of this thesis is organized as follows: In Chapter II we formulate two LP models, representing alternative approaches to traffic engineering discussed above. In Chapter III we use these models to perform a detailed analysis of Abilene, the backbone for the Internet2 academic network. Finally in Chapter IV we summarize the contributions of the thesis and offer suggestions for future research.

THIS PAGE INTENTIONALLY LEFT BLANK

II. MODEL FORMULATION

Because IP flows are defined essentially in terms of Source (node *s*) and Terminal (node *t*) node pairs, we represent IP traffic engineering as a multi-commodity flow problem in which each *s-t* pair is treated as a separate commodity. In our network, the "edge" nodes (i.e., the nodes the provide network access to the users and connect to the internal nodes) represent the users. We will assume that all users communicate with one another, and that the demand for flow between user pairs is proportional to the product of their capacities, an assumption consistent with the aforementioned gravity model of traffic demand. The "internal" nodes (i.e., nodes that provide connectivity to the other network devices) represent intra-network routing devices (i.e., "routers"), and arcs connecting them represent one-hop IP connectivity between routers (i.e., routers directly "see" one another according to IP).

The primary problem is to identify the maximum flow (and corresponding optimal routing) for a multi-commodity network that satisfies the proportionality constraints for flow demand as well as capacity constraints on nodes and arcs.

The secondary problem is to identify the optimal interdiction of those flows that reduces that maximum flow in the worst possible way.

We consider two approaches to traffic engineering. First, we consider a strict approach where each commodity follows a single shortest path. Second, we will look at best case traffic engineering in which it is possible to route traffic through the network by taking multiple, possibly longer, paths.

A. SINGLE PATH MULTI-COMMODITY MAXIMUM FLOW

1. Solving for Maximum Flow

This model represents the simplest form of traffic engineering. Traffic from user *s* to user *t* follows a single path in the network. That path is the computed shortest path, in terms of the number of internal nodes visited (or arcs traversed), from user *s*, to user *t*. The total amount of data flow through the network is the sum of the traffic routed along all of the shortest *s*-*t* paths. There exist flow throughput capacities on the network’s “internal” nodes and/or the arcs connecting them. We desire to maximize this total amount of data flow by raising flow along every path in a proportional manner until one of the internal nodes and/or connecting arcs reaches capacity. We refer to the network components that reach capacity as "bottlenecks."

Formulation 1: MAX SP (Maximizing Single-Path Flow)

Index Use

$i, j, k \in N$	Nodes
$(i, j) \in A$	Directed arc from node *i* to node *j*
$s, t \in E \subseteq N$	Source and terminal nodes in the set of “edge” nodes E

Data

D_s	Traffic demand by edge node $s \in E$ [flow]
B_k	Throughput capacity of node $k \in N$ [flow]
$u_{i,j}$	Upper bound on flow from node *i* to node *j* for each arc $(i, j) \in A$ [flow]
$r^{s,t}$	Shortest path route from node *s* to node *t* for each $s, t \in E$ [flow]

Calculated Data

$r_k^{s,t}$

Binary indicator whether node *k* is on the shortest path from node *s* to node *t* for *s*, *t* $\in$ *E* [binary]

$$r_k^{s,t} = \left(\begin{array}{l} 1 \quad \text{if node k is on } r^{s,t} \\ 0 \quad \text{if node k is not on } r^{s,t} \end{array} \right.$$

$q_{i,j}^{s,t}$

Binary indicator whether arc (*i*, *j*) is on the shortest path from node *s* to node *t* for *s*, *t* $\in$ *E* [binary]

$$q_{i,j}^{s,t} = \left(\begin{array}{l} 1 \text{ if arc}(i,\, j) \text{is on } r^{s,t} \\ 0 \text{ if arc}(i,\, j) \text{is not on } r^{s,t} \end{array} \right.$$

Decision Variable

$X^{s,t}$

Flow along route $r^{s,t}$ from node *s* to node *t* [flow]

$X^{s,t} = \rho B_s B_t$ where *ρ* is a constant of proportionality

Formulation

$$\max_{\rho} \quad \sum_{s,\, t \in E} X^{s,t} \tag{C1.0}$$

$$\text{s.t.} \quad \sum_{s,\, t \in E} X^{s,t} r_k^{s,t} \leq B_k \qquad \forall k \in N \tag{C1.1}$$

$$\sum_{s,\, t \in E} X^{s,t} q_{i,j}^{s,t} \leq u_{i,j} \qquad \forall (i,j) \in A \tag{C1.2}$$

$$X^{s,t} = \rho B_s B_t \qquad \forall (s,t) \in E \times E \tag{C1.3}$$

(NOTE: Throughout the thesis, n denotes the number of nodes, e denotes the number of edge nodes, and m denotes the number of arcs.)

Discussion

Equation (C1.0) is the objective function which represents total amount of data flow through the network. It is the sum of the traffic routed along all of the shortest *s-t* paths. We maximize the objective function value by increasing the proportionality constant *ρ*. Equations (C1.1) and (C1.2) limit the amount of flow through each node and arc respectively. Equation (C1.3) ensures that flow is routed between each *s-t* pair, and that those flows are raised in proportion to each other.

There is considerable preprocessing involved in solving for single-path maximum flow. We compute the $r^{s,t}$ values using the Floyd-Warshall algorithm (Appendix A). Floyd-Warshall determines the shortest paths between all node pairs, but we are only interested the shortest paths between each *s-t* node pair.

Once the shortest path routes are determined, we use it to build a matrix

(size: $e^2 \times n$) of $r_k^{s,t}$ values $\begin{array}{cc} & k \longrightarrow \in N \\ (s,t) & \begin{bmatrix} \quad r_k^{s,t} \quad \end{bmatrix} \end{array}$ (Appendix B) and a matrix (size : $e^2 \times m$) of $q_{i,j}^{s,t}$ values

$$\begin{array}{cc} & (i,j) \longrightarrow \in A \\ (s,t) & \begin{bmatrix} \quad & & \quad \\ & q_{i,j}^{s,t} & \\ & & \end{bmatrix} \end{array}$$

(Appendix C).

Each of the three tasks mentioned above runs in O(n^3).

Our formulation allows us to study networks in which only the nodes are capacitated ($B_k < \infty$, $u_{i,j} = \infty$), or when just the arcs are capacitated ($B_k = \infty$, $u_{i,j} < \infty$), or when both nodes and arcs are capacitated ($B_k < \infty$, $u_{i,j} < \infty$).

The special structure associated with the constant of proportionality ρ affords a direct analytic solution to the maximum flow under single-path routing. For a network with capacitated nodes and un-capacitated arcs, consider equations (C1.1) and (C1.3).

$$\sum_{s,\,t \in E} X^{s,t} r_k^{s,t} \le B_k \qquad \forall k \in N \qquad \text{(C1.1)}$$

$$X^{s,t} = \rho B_s B_t \qquad \forall (s,t) \in E \times E \qquad \text{(C1.3)}$$

Equation (C1.1) can be rewritten as

$$\sum_{s,\,t} \rho B_s B_t \; r_k^{s,t} \le B_k \qquad \forall k \in N$$

or

$$\rho \le \frac{B_k}{\sum_{s,\,t} r_k^{s,t} \, B_s B_t} \qquad \forall k \in N$$

Now we can solve for ρ directly.

$$\rho = \min_k \left(\frac{B_k}{\sum_{s,\,t} r_k^{s,t} \, B_s B_t} \right) > 0 \qquad \forall k \in N \qquad (1)$$

For a network with capacitated arcs and un-capacitated nodes, we solve for ρ using equations (C1.2) and (C1.3) and performing the same substitution. Solving for ρ in this type of network yields the following result

$$\rho \;=\min_{(i,j)\in A}\left(\frac{u_{i,j}}{\sum_{s,t} q_{i,j}^{s,t}\, B_s B_t}\right) > 0 \qquad \forall (i,j)\in A \quad (2)$$

For networks where both the nodes and arcs have capacity, the correct ρ is the minimum ρ between equations (1) and (2). This type of solution is easily implemented in a spreadsheet program such as EXCEL.

2. Minimizing the Maximum Flow

Suppose an opponent (an *attacker*) wants to incur the greatest amount of "damage" on the network. Assume that the *attacker* has the capability to destroy a limited number of nodes and/or arcs, thus reducing to zero the capacity for each of the destroyed nodes and/or arcs (B_k=0 and/or $u_{i,j}$=0). The *attacker* must decide which nodes and/or arcs in the network to destroy so that the maximum flow is minimized, perhaps to zero.

The previous formulation is the same, with the addition of the following data and decision variables.

Formulation 2: MIN-MAX SP (Minimizing the maximizing Single-Path Flow)

<u>Data</u>

$attacks$	Number of nodes and/or arcs that the attacker can destroy [cardinality]

Decision Variable

Y_k Binary indicator for attacker destruction of node

k ∈ *N* [binary]

$$Y_k = \begin{cases} 1 \ \text{ if } \mathit{node}\, k \,\text{is destroyed} \\ 0 \ \text{otherwise} \end{cases}$$

$Y_{i,j}$ Binary indicator for attacker destruction of

arc (*i*, *j*) ∈ *A* [binary]

$$Y_{i,j} = \begin{cases} 1 \ \text{ if } \mathit{arc}(i,j)\,\text{is destroyed} \\ 0 \ \text{otherwise} \end{cases}$$

Min-Max optimization of flow

$$\min_{Y\in\Upsilon}\left\{\begin{array}{llll} \max\limits_{\rho} & \sum\limits_{s,\,t\in E} X^{s,t} & & \text{(C2.0)} \\ & & & \\ s.t. & \sum\limits_{s,\,t\in E} X^{s,t} r_k^{s,t} \le B_k(1-Y_k) & \forall k\in N & \text{(C2.1)} \\ & & & \\ & \sum\limits_{s,\,t\in E} X^{s,t} q_{i,j}^{s,t} \le u_{i,j}(1-Y_{i,j}) & \forall (i,j)\in A & \text{(C2.2)} \\ & & & \\ & X^{s,t} = \rho B_s(1-Y_s) B_t(1-Y_t) & \forall \left(s,t\right)\in E\times E & \text{(C2.3)} \end{array}\right\}$$

$$\text{where } Y\in\Upsilon = \left\{\begin{array}{ll} \sum\limits_{k\in N} Y_k + \frac{1}{2}\sum\limits_{(i,\,j)\in A} Y_{i,j} \ \le \mathit{attacks} & \text{(C2.4)} \\ Y_{i,j} = Y_{j,i} & \text{(C2.5)} \\ Y_k,\, Y_{i,j} \in \left\{0,1\right\} \quad \forall\, i,\ j,\ k\in N & \end{array}\right\}$$

Discussion

Equation (C2.0), the objective function, reflects that the *attacker* desires to minimize the previously maximized sum of traffic routed along all of the shortest *s-t* paths in the network. The *attacker* will seek to destroy nodes, or arcs, or both depending on the network's structure (i.e., which components are capacitated). Equations (C2.1) and (C2.2) limit the amount of flow through each node and arc respectively. Equation (C2.3) ensures that flow is routed between each *s-t* pair, and that those flows are raised in proportion to each other. Equation (C2.4) places a limit of the number of attacks that the *attacker* can prosecute. Destroying a node or arc drops its capacity to zero. Equation (C2.5) states that destroying arc (*i*, *j*) also destroys arc (*j*, *i*).

Solving by Total Enumeration

There are a finite number of Y_k and $Y_{i,j}$ variables for the model. Thus, the optimal "interdiction" solution can be determined by checking all possible choices for Y_k and $Y_{i,j}$ for a given value of $attacks$, and then keeping the "best" solution (i.e., the solution that minimizes the maximum flow through the network the most, as in Rardin et al., 1998).

This type of total enumeration works for problems with limited size. For the network operator (the *defender*), there are $e(e$-1) decision variables, one for each *s-t* pair. There are n+m decision variables for the attacker.

Discussion

When a node or arc is attacked, it is removed from the network. We then use the Floyd-Warshall algorithm to re-compute the shortest paths for the remaining *s-t* pairs so that the new $r_k^{s,t}$ and $q_{i,j}^{s,t}$ values can be can calculated.

Changing the $r_k^{s,t}$ and $q_{i,j}^{s,t}$ values directly impacts flow through the network, as measured by *ρ*. In some instances *ρ* will decease, as expected. However, in other instances *ρ* may actually increase. This is the converse of

Braess's paradox, which states that adding additional capacity to a network can reduce the network's total flow (see Florian and Hearn 1995). In our case it is possible that, by attacking certain nodes or arcs, bottlenecks are removed from the network resulting in a net *increase* in flow through the remaining network.

B. MULTIPLE PATH MULTI-COMMODITY MAXIMUM FLOW MODEL

1. Solving for the Maximum Flow

This model represents best-case traffic engineering in that the network is able to route traffic along multiple, possibly longer, paths, and makes better overall use of network resources. Here, we modify the standard LP formulation of the Maximum *s-t* Flow problem (Appendix D) to accommodate multi-commodity flows while also adding a proportionality constraint for each *s-t* pair. We use the technique of "node splitting" to replace the capacity of a node with a capacitated arc connecting the two split nodes. In this manner, all capacities are represented as arc capacities. Like the single-path model, the goal is to maximize that total amount of data flow through the network. We introduce an artificial return arc for every *s-t* pair. The return arcs are unbounded ($u_{s,t} = \infty$), but must adhere to the constant of proportionality. We maximize the sum of the flows on the return arcs, again, by increasing flow along all of them in proportion to each other until one of the arcs in the network reaches its capacity.

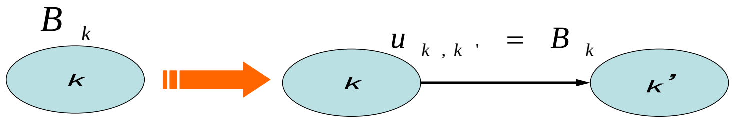
Formulation 3: MAX MP (Maximizing Multiple-Path Flow)

<u>Index Use</u>

$i,\ j,\ k \in N$	Nodes
$(i,\ j) \in A$	Directed arc from node *i* to node *j*
$s,t \in E \subseteq N$	Source and terminal nodes in the set of "edge" nodes *E*

Preprocessing

Each "internal" node $k \in E$ is split into two nodes {*k*, *k'*} with directed arc (*k*, *k'*) connecting them.



Data

B_k Throughput capacity at node $k \in N$ [flow]

D_s Demand for edge node $s \in E$ [flow]

$u_{i,j}$ Upper bound on flow from node *i* to node *j* on arc $(i, j) \in A$ [flow]

$$u_{i,j} = \begin{cases} B_k & \text{if } i = k,\ j = k' \\ \infty & \text{otherwise} \end{cases}$$

Decision Variables

$X_{i,j}^{s,t}$ "Internal" flow of commodity *s*-*t* on arc $(i, j) \in A$ [flow]

$Z^{s,t}$ "Return" flow of commodity *s*-*t* on artificial arc $(t\,,s) \in A$ [flow]

$Z^{s,t} = \rho D_s D_t$ where ρ is a constant of proportionality.

<u>Formulation</u> [dual variables]

$$\max_{\rho} \sum_{s,\,t \in E} Z^{s,\,t} = \max_{\rho} \rho \sum_{s,\,t \in E} D_s D_t \qquad \text{(C3.0)}$$

$$\text{s.t} \sum_{(k,\,j)\in A} X^{s,\,t}_{k,\,j} - \sum_{(i,\,k)\in A} X^{s,\,t}_{i,\,k} = \begin{cases} Z^{s,\,t} & \text{if } \; k = s \\ 0 & \text{if } \; k \neq s,t \;\; \forall k \in N, \forall (s,t) \in E \times E \quad [\alpha^{s,\,t}_{k}] \;\; \text{(C3.1)} \\ -Z^{s,\,t} & \text{if } \; k = t \end{cases}$$

$$\sum_{s,\,t \in E} X^{s,\,t}_{i,\,j} \leq u_{i,\,j} \qquad \forall\, (i,\,j) \in A \qquad [\,\beta_{i,\,j}]\;\text{(C3.2)}$$

$$Z^{s,\,t} - \rho D_s D_t = 0 \qquad \forall\, (s,t) \in E \times E \qquad [\mu^{s,\,t}] \;\; \text{(C3.4)}$$

$$X^{s,t}_{i,\,j} \geq 0,\; Z^{s,t} \geq 0, \qquad \rho \;\; U.R.S.$$

<u>Discussion</u>

Equation (C3.0), the objective function, represents the sum of the flows along the return arcs (*t, s*). We maximize the objective function value by increasing the proportionality constant *ρ*. Equation (C3.2) is a balance of flow constraint. Equation (C3.2) limits the amount of flow on each arc. Equation (C3.4) ensures that flow is routed along each return arc (*t*, *s*), and that those flows are raised in proportion to each other.

The following table shows the number of decision variables and constraints contained in multiple-path model:

Decision Variables		**Constraints**	
Flow from *s* to *t*	$e \cdot (e\text{-}1)$	Flow Balance	$n \cdot e \cdot (e\text{-}1)$
Arc Flow	$m \cdot e \cdot (e\text{-}1)$	Arc Capacity	m
		Demand	$e \cdot (e\text{-}1)$

The multiple-path model is a linear programming formulation that we solve using General Algebraic Modeling System (GAMS) software and the Solver CPLEX. The effort GAMS requires to solve the multiple-path model grows significantly with the number of *s-t* pairs, $e(e\text{-}1)$, in a given network.

2. Minimizing the Maximum Flow

Consider again the case of an *attacker* who can disable a finite number of network components and seeks to damage the total network flow in the worst possible manner. A natural choice to represent the effect of an arc attack is to set the capacity of the attacked arc to zero (as was done in the single-path model). However, an equivalent and computationally attractive approach is to assign a penalty cost, $v_{i,j}$, to attacked arcs. This discourages the *defender* from sending flow across an arc that's been destroyed. To avoid attacked arcs, the penalty cost must be greater the one, because, if $v_{i,j}$=1 the *defender* is completely indifferent to sending flow across the interdicted arc, and the resulting problem may have many equivalent optimal solutions. Thus we set $v_{i,j}$=2 if arc (*i, j*) is susceptible to being attacked. We can similarly designate an arc as invulnerable by setting $v_{i,j}$=0. In this model, artificial return arcs are all invulnerable.

The previous formulation is the same, with the addition of the following data and decision variables.

Formulation 4: MIN-MAX MP (Minimizing the Maximizing Multiple-Path Flow)

<u>Data</u>

$v_{i,j}$	Penalty cost for arc (*i*, *j*) $\in$ *A* [cost/flow]
attacks	Number of arcs the attacker can destroy [cardinality]

<u>Decision Variables</u>

$Y_{(i,\,j)}$	Attacker destruction of arc (*i*, *j*) $\in$ *A* [binary]

$$Y_{i,j} = \begin{cases} 1 \ \text{ if } arc(i,j) \,\text{is destroyed} \\ 0 \ \text{ otherwise} \end{cases}$$

Min-Max optimization of flow[dual variables]

$$\min_{Y\in\Upsilon}\left\{\begin{array}{llll} \max\limits_{\rho} \sum\limits_{s,t\in E}\left(Z^{s,t} - \sum\limits_{(i,j)\in A} v_{i,j}\, X^{s,t}_{i,j}\, Y_{i,j}\right) & & & (\text{C4.0}) \\ \text{s.t} \quad \sum\limits_{(k,j)\in A} X^{s,t}_{k,j} - \sum\limits_{(i,k)\in A} X^{s,t}_{i,k} = \begin{cases} Z^{s,t} & \text{if } k=s \\ 0 & \text{if } k\neq s,t \ \ \forall k\in N, \forall(s,t)\in E\times E \\ -Z^{s,t} & \text{if } k=t \end{cases} & & [\alpha^{s,t}_{k}] & (\text{C4.1}) \\ \qquad \sum\limits_{s,t\in E} X^{s,t}_{i,j} \le u_{i,j} & \forall(i,j)\in A & [\beta_{i,j}] & (\text{C4.2}) \\ \qquad Z^{s,t} - \rho D_s D_t = 0 & \forall(s,t)\in E\times E & [\mu^{s,t}] & (\text{C4.3}) \\ \qquad X^{s,t}_{i,j} \ge 0,\ Z^{s,t} \ge 0, \qquad \rho\ U.R.S. & & & \end{array}\right\}$$

$$\text{where } Y\in\Upsilon = \left\{\begin{array}{ll} \dfrac{1}{2}\sum\limits_{(i,j)\in A} Y_{i,j} + \sum\limits_{k\in N} Y_{k,k'} \le \mathit{attacks} & (\text{C4.4}) \\ Y_{i,j} = Y_{j,i} & (\text{C4.5}) \\ Y_{i,j}, Y_{k,k'} \in \{0,1\} \qquad \forall(i,j)\in A & \end{array}\right\}$$

Taking the dual of the min-max formulation yields the following formulation.

Formulation 5: MAX MP Dual (Minimizing the Maximizing Multiple-Path Flow)

Min-Max optimization of flow

$$\min_{\alpha,\beta,\mu,Y} \sum_{(i,j)\in A} u_{i,j}\beta_{i,j} \qquad \text{(C5.0)}$$

$$s.t. \quad \alpha_i^{s,t} - \alpha_j^{s,t} + \beta_{i,j} + v_{i,j}Y_{i,j} \ge 0 \qquad \forall (i,j)\in A, \forall s,t \in N \quad \left[X_{i,j}^{s,t}\right] \text{(C5.1)}$$

$$\alpha_t^{s,t} - \alpha_s^{s,t} + \mu^{s,t} \ge 1 \qquad \forall s,t \in N \qquad \left[Z^{s,t}\right] \text{ (C5.2)}$$

$$\sum_{s,t\in E} D_s D_t \mu^{s,t} = 0 \qquad \text{(C5.3)}$$

$$\alpha_i^{s,t} \text{ U.R.S.,} \quad \mu^{s,t} \text{ U.R.S.} \qquad \forall\,(s,t)\in E\times E$$

$$\beta_{i,j} \ge 0 \qquad \forall (i,j)\in A$$

$$\sum_{(i,j)\in A} Y_{i,j} \le attacks \qquad \text{(C5.4)}$$

$$Y_{i,j} \in \{0,1\} \qquad \forall (i,j)\in A$$

Discussion

This formulation is the dual of the maximizing multiple-path flow formulation. The formulation consists of dual variables for flow balance: equation (C5.1), arc capacity: equation (C5.2), and demand: equation (C5.3) for each *s-t* pair. Like the single-path model, the *attacker* desires to minimize the previously maximized sum of traffic routed along all return arcs (*t*, *s*) in the network.

The following table shows the number of decision variables and constraints contained in multiple-path MIP model (dual):

Dual Decision Variables		Dual Constraints	
Node Flow	$n \cdot e \cdot (e$-1)	Flow Balance	$m \cdot e \cdot (e$-1)
Upper bound	m	Arc Capacity	$e \cdot (e$-1)
Commodity Flow	$e \cdot (e$-1)	Demand	$e \cdot (e$-1)

The multiple-path dual model is a mixed integer program that we also solve using GAMS and the CPLEX Solver. Again, the time GAMS requires to solve this problem grows significantly with the number of *s-t* pairs in the network.

Working with Proportional Flow

If a user is disconnected from the network as a result of an attack, flow to or from that user is no longer possible. Thus, that user cannot send or receive traffic, so $Z^{s,t}$=0. Just like in the single-path formulation, flows are constrained to be proportional to each other ($Z^{s,t} - \rho D_s D_t = 0$), so the disconnection of a single edge node from the network effectively sets *ρ*=0 and all flows disappear. In practice, the disconnection of a user does not preclude other users from sending and/or receiving traffic. In this sense, the proportionality constraint used here is unrealistic. In order to facilitate the computation of a more reasonable traffic response to an attacked network, we consider the following model re-formulation:

Let $\rho^{s,t}$ be the proportionality constant for a single *s-t* pair. We modify the equation (C3.4)

$$Z^{s,t} - \rho^{s,t} D_s D_t = 0 \qquad \forall (s, t) \in N$$

And add an additional constraint

$$\rho^{s,t} = \rho R^{s,t} \qquad \forall (s, t) \in N$$

Where $R^{s,t}$ =1 if there exists a path connecting node *s* to node *t*, or $R^{s,t}$ =0 if no such path exists. Thus $R^{s,t}$ is a binary value that indicates whether an individual *s*-*t* path is available in the network.

There are two approaches for determining the $R^{s,t}$ values. The first is to let them be binary variables and have the model determine the best choices. (This makes the primal problem a MIP.) A drawback with this approach is that while the model will never allow $R^{s,t} = 1$ if *s* and *t* are not connected, *s*-*t* pairs that are connected might also be shut off proactively in order to provide a better solution for maximizing flow through the remaining network (again the Braess Paradox). Such a solution is contrary to the "fairness" assumption underlying our use of proportional flows.

An alternative approach is to pre-compute the $R^{s,t}$ values using a reachability algorithm (Appendix E). The multiple-path model (Figure 3) remains the same, with the addition of the following data.

Formulation 6: R-MAX MP (Revised Maximizing Multiple-Path Flow)

<u>Calculated Data</u>

$R^{s,t}$ Connection between node *s* and node *t* [binary]

$$R^{s,t} = \begin{cases} 1 & \text{if node } t \text{ is "reachable" from node } s \\ 0 & \text{if node } t \text{ is not "reachable" from node } s \end{cases}$$

Formulation

$$\max_{\rho} \sum_{s,\,t \in E} Z^{s,\,t} = \max_{\rho} \rho \sum_{s,\,t \in E} D_s D_t \qquad \text{(C6.0)}$$

$$\text{s.t} \sum_{(k,\,j) \in A} X^{s,\,t}_{k,\,j} - \sum_{(i,\,k) \in A} X^{s,\,t}_{i,\,k} = \begin{cases} Z^{s,\,t} & \text{if } k = s \\ 0 & \text{if } k \neq s,t \ \forall k \in N, \forall (s,t) \in E \times E \\ -Z^{s,\,t} & \text{if } k = t \end{cases} \qquad \text{(C6.1)}$$

$$\sum_{s,\,t \in E} X^{s,\,t}_{i,\,j} \leq u_{i,\,j} \qquad \forall (i,\,j) \in A \qquad \text{(C6.2)}$$

$$Z^{s,\,t} - \rho^{s,\,t} D_s D_t = 0 \qquad \forall (s,\,t) \in E \times E \quad \text{(C6.3)}$$

$$\rho^{s,\,t} = \rho R^{s,t} \qquad \forall (s,\,t) \in E \times E \quad \text{(C6.4)}$$

$$\rho \ U.R.S.$$

$$X^{s,t}_{i,j} \geq 0, \quad Z^{s,t} \geq 0, \quad R^{s,t} \in \{0,1\} \qquad \forall (s,\,t) \in E \times E$$

Discussion

In summary, MAX MP calculates total flow through the network under best-case traffic engineering. MAX MP Dual determines the optimal arc(s) to attack in order to reduce flow through the network the most. And finally, R-MAX MP calculates total flow on a “damaged” network. Just like for the single-path model, this model allows us to study networks in which only the nodes are capacitated ($u_{i,\,j} = \infty$), or when just the arcs are capacitated ($B_k = \infty$), or when both nodes and arcs are capacitated.

THIS PAGE INTENTIONALLY LEFT BLANK

III. ANALYZING THE ABILENE NETWORK

Abilene is the high-speed backbone of the Internet2 educational network, a not-for-profit advanced networking consortium of universities, laboratories, and government agencies. (Detailed information is available at http://www.internet2.edu/) Figure 1 represents Abilene's network topology (as of 2004). We use the models developed in the previous chapter to examine how different methods of traffic engineering affect the carrying capacity of the network, as defined by its multi-commodity maximum flow.

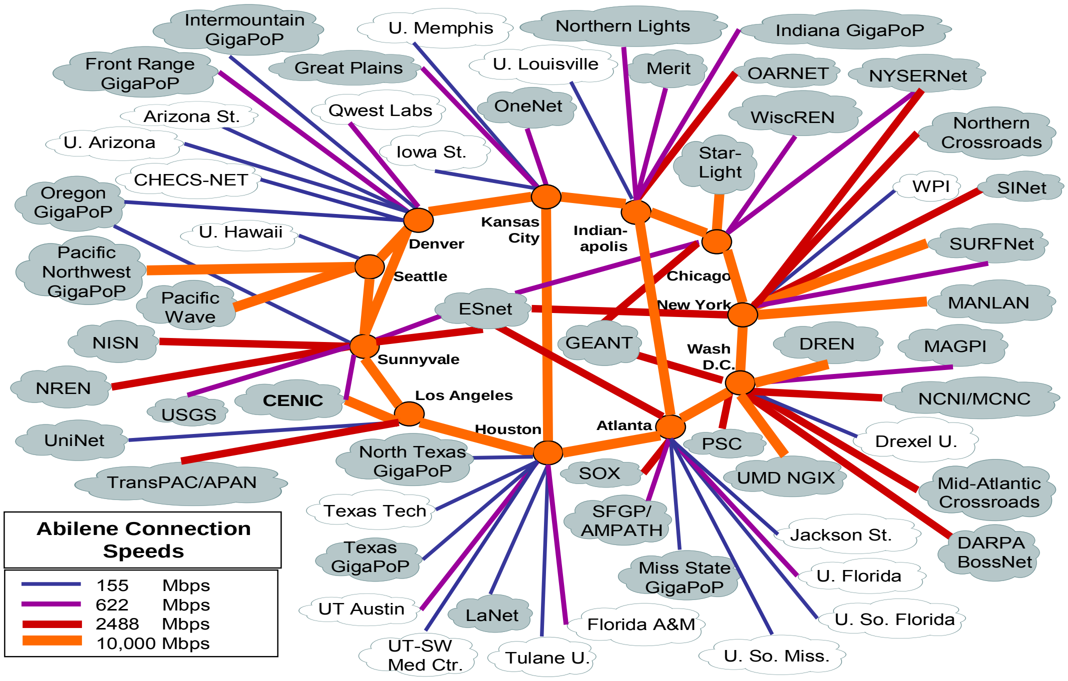


Figure 1. The Abilene Network

- The clouds in the figure are customers, either campus networks (white) or other network providers (grey).
- Abilene has a total of 58 customers and/or peers. CENIC, ESnet, GEANT, NYSERNet, and Oregon GigaPoP all use Abilene at more the one location. We treat each of those connections as multiple customers, bringing the total number to 65.
- There are 4,160 customer-to-customer pairs ($e \cdot (e-1) = 65 \cdot 64 = 4160$).
- Each of the eleven circles represents a transshipment node, specifically a Juniper T640 Router, located in a major U.S. city.
- The arcs are undirected with line colors and thickness indicating traffic capacity (i.e., bandwidth), which we use as a proxy for customer demand for traffic (i.e., the demand for customer s to route traffic to customer t is equivalent to the product of their bandwidth capacity).
- There exist fourteen, two-way connections amongst the transshipment routers.

As detailed in the previous chapter, we seek to maximize the amount of traffic carried among the 4,160 customer-to-customer pairs. The MAX SP represents the simplest form of traffic engineering in which data is routed between customers via the single “shortest” path as seen by IP. The MAX MP represents the best-case scenario for traffic engineering, in which data sent from customer to customer can be split into multiple streams, each following its own path. Sometimes the optimal multiple paths are longer than the shortest path, sometimes they are the same. Both formulations raise all 4,160 flows in the network in proportion to one another (via the constant of proportionality ρ) until at least one of the network components reaches capacity and becomes saturated.

A. OPTIMAL FLOWS WITH NODE CAPACITIES

In practice, both nodes and arcs are capacitated, but here we will focus on the throughput capacity of transshipment routers in the network. Here the transshipment routers (the nodes) each have a maximum capacity of 320,000 megabits per second (Mbps), which represents the highest combination of line cards supported by the T640 Router at the time the data was collected.

1. Maximum Flow through Abilene

The total amount of traffic routed between customers using MAX SP and MAX MP is 630,941 Mbps (C1.0) and 738,442 Mbps (C3.0) respectively. Those results, along with the transshipment node utilizations are displayed in the table below.

Utilization of an internal node (router) is simply the percentage of that router's maximum capacity used by the 4,160 customer-to-customer pairs (MAX SP: $\frac{\sum_{s,t} q_{s,t}^{i,j}}{B_k}$, MAX MP: $\frac{\sum_{s,t} X_{i,j}^{s,t}}{u_{i,j}}$).

NOTE: The units on the "flow" values in the tables and figures throughout this chapter are in Mbps.

Table 1. Utilization of Abilene Transshipment Routers Under Maximum Flows

	MAX SP	MAX MP	Increase
Total-Flow	630,941	738,442	15%
ρ	0.000035	0.000041	15%
ATLANTA	0.742	1	26%
CHICAGO	0.652	0.940	29%
DENVER	0.578	0.887	31%
HOUSTON	0.608	0.963	36%
INDIANAPOLIS	0.528	0.927	40%
KANSAS CITY	0.595	1	41%
LOS ANGELES	0.439	0.469	3%
NEW YORK	0.901	0.939	4%
SEATTLE	0.541	0.633	9%
SUNNYVALE	0.335	0.395	6%

WASHINGTON DC	1	1	0%

The less restrictive MAX MP achieves a 15% increase of flow through the network. Also, every transshipment router is utilized more than it is in MAX SP, and the increases vary by router.

In both models, the Washington D.C. router is the first to reach its capacity (MAX SP: $\frac{\sum_{s,t} q_{i,j}^{s,t}}{B_k}$=1, MAX MP: $\frac{\sum_{s,t} X_{i,j}^{s,t}}{u_{i,j}}$=1) and thus is the “bottleneck.” It is preventing a further increase in flow (ρ). In the multiple-path model, the Atlanta and Kansas City routers are also saturated.

Increasing the throughput capacity of the bottleneck router(s) in the network would enable an increase in flow through the network. For example, if we could double the capacity of Washington D.C. ($B_{Washington\,DC}$ =640,000 Mbps, perhaps by operating two Juniper T640 routers in parallel) we would increase flow 10% (ρ=.000039) for single-path routing. Doing the same to Sunnyvale instead produces a 0% flow increase.

In practice, it may not be feasible, or necessary, to increase the capacity of every transshipment router in order to improve total throughput, thus identifying the bottleneck(s) is significant.

The next two figures demonstrate the actual data flows between the transshipment routers. The bold italicized number adjacent to the router is the sum of the demands of the customers located at that particular router. The number in each node is its utilization (expressed as a fraction of its capacity) under maximum flow conditions. These numbers correspond to the values in Table 1. Not shown is the data flow between customers who use Abilene at the same transshipment router.

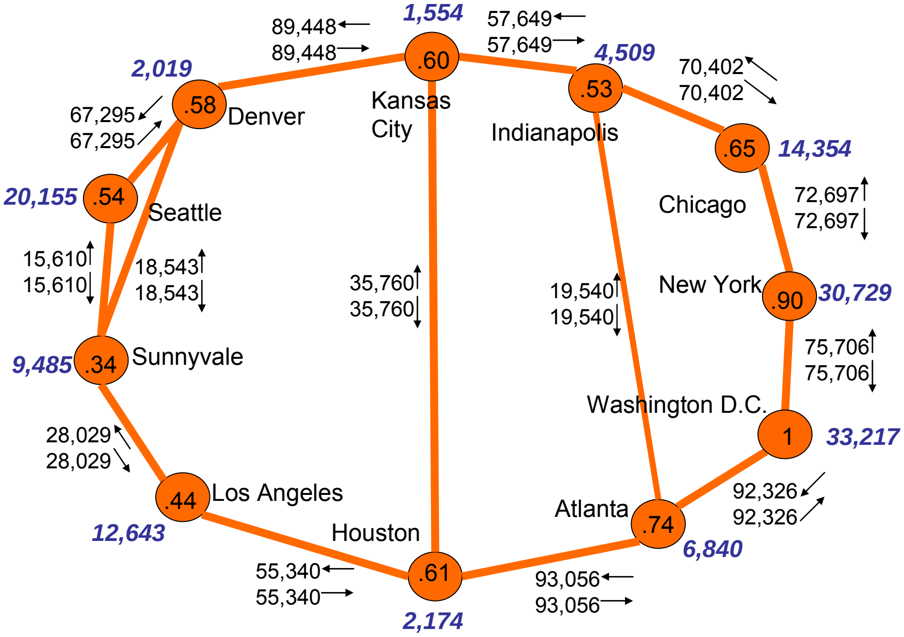


Figure 2. Abilene Single-Path Flow through Nodes

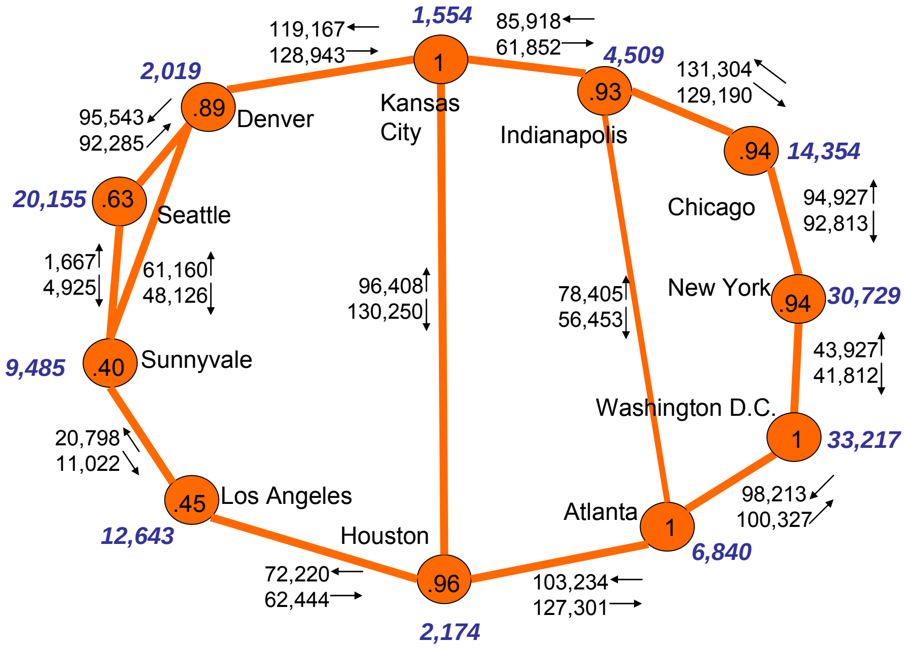


Figure 3. Abilene Multiple-Path Flow through Nodes

As expected, the flow levels in MAX SP (Figure 2) are symmetric since the shortest paths between the transshipment routers are also symmetric.

In Figure 3, the flow levels on each arc are no longer symmetric since multiple-path model uses all available capacity, even if not on the shortest path.

The difference in the flow values between the two figures on the transshipment connections can be explained by the MAX MP's ability use longer and/or multiple routes for sending traffic between customers.

The next two figures show the paths for data destined to the New York (dashed green arrows) and Sunnyvale (solid blue arrows) routers from the other routers in the network (MAX SP: $q_{i,j}^{s,t}$, MAX MP: $X_{i,j}^{s,t}$ where t=New York and Sunnyvale).

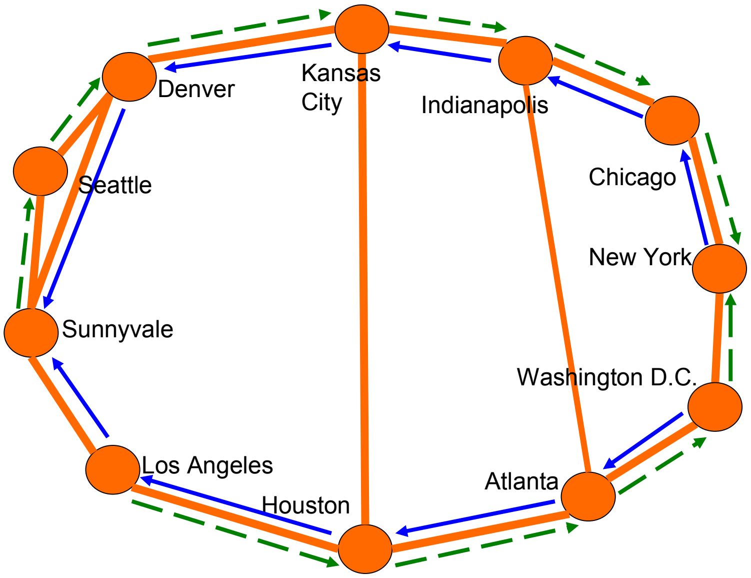


Figure 4.   Single-Path Flow to New York & Sunnyvale Routers

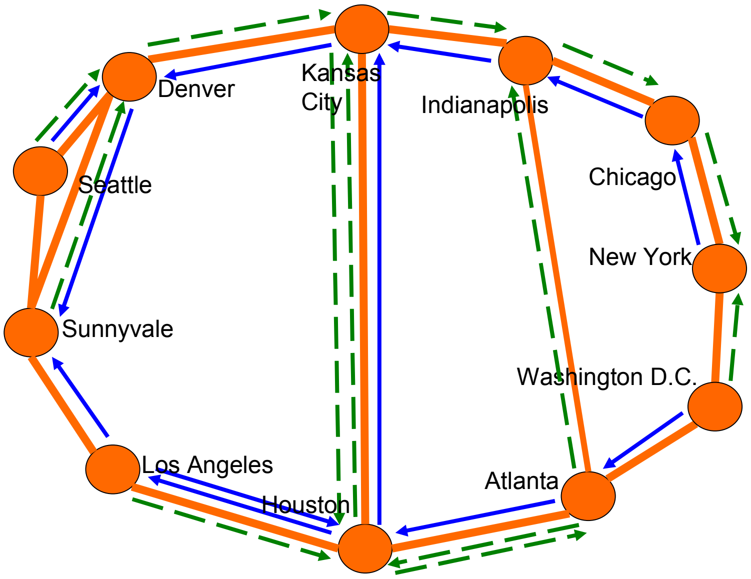


Figure 5.   Multiple-Path Flow to New York & Sunnyvale Routers

The total amounts of traffic traveling to New York and Sunnyvale are shown in Table 2 (MAX SP: $\sum_{s} X^{s,t}$, MAX MP: $\sum_{s} Z^{s,t}$ where t=New York and Sunnyvale). The numbers shown simply reflect the differences in *ρ* values between the single-path and multiple-path solutions.

Table 2. Traffic Flow to New York & Sunnyvale Routers

		Single-Path		Multiple-Path	
		To NEW YORK	To SUNNYVALE	To NEW YORK	To SUNNYVALE
From	ATLANTA	7,357	2,271	8,610	2,658
	CHICAGO	15,438	4,765	18,068	5,577
	DENVER	2,171	670	2,541	784
	HOUSTON	2,338	722	2,737	845
	INDIANAPOLIS	4,850	1,497	5,676	1,752
	KANSAS CITY	1,671	516	1,956	604
	LOS ANGELES	13,598	4,197	15,915	4,912
	NEW YORK	------	10,201	------	11,939
	SEATTLE	21,677	6,691	25,370	7,831
	SUNNYVALE	10,201	------	11,939	------
	WASHINGTON DC	35,725	11,027	41,812	12,906

Abilene data reduction

Both the single-path (EXCEL) and multiple-path (GAMS) models take a considerable amount of time to the execute Abilene data. 4,160 customer-to-customer paths translates into a large number of decision variables and constraints. An example of this is shown in the table below.

Single-Path (Figure 1)			TOTAL	
Decision Variables	Customer-to-Customer Pairs	4,160	**Variables**	4,160
Constraints	Router Capacity	11	**Constraints**	4,329
	Arc Capacity	158		
	Flow Proportionality	4,160		
Multiple-Path (Figure 3)				
Decision Variables	Flow on Return Arcs	4,160	**Variables**	4,329
	Flow through Nodes	169		
Constraints	Balance of Flow	361,920	**Constraints**	366,249
	Arc Capacity	169		
	Flow Proportionality	4,160		

We can significantly reduce the number of customer-to-customer paths if we only consider paths between the eleven transshipment routers. The demand (B_k) at each router can be aggregated from the sum of the demands of that router's customers (same values in Figure 2 and Figure 3). However, this reduction does not account for the traffic routed between customers who use the same router. So we leave those paths in our data reduction (i.e., for example, keep the paths from University of Hawaii to Pacific Northwest GigaPoP and to Pacific Wave, but get rid of the paths to sixty-two paths).

We reduce the number of customer-to-customer paths to 462 (110 router-to-router paths plus 362 total "local" customer paths).

Single-Path (Figure 1)			TOTAL	
Decision Variables	Customer-to-Customer Pairs	462	**Variables**	4,160
Constraints	Router Capacity	11	**Constraints**	631
	Arc Capacity	158		
	Flow Proportionality	462		
Multiple-Path (Figure 3)				
Decision Variables	Flow on Return Arcs	462	**Variables**	631
	Flow through Nodes	169		
Constraints	Balance of Flow	40,194	**Constraints**	40,825
	Arc Capacity	169		
	Flow Proportionality	462		

This reduction dramatically improves the model run times, from minutes to seconds.

2. Single Node Attack

Now we consider the impact of losing one of the transshipment routers. Causes for a losing a router range from equipment failure to a deliberate attack. When a router is lost, its throughput capacity goes to zero (B_k=0) making it unavailable to the network. Thus customers connected to that router are no longer able to send and receive traffic from the other customers.

The top five “optimal” router attacks obtained via enumeration (here, $\binom{11}{1}$ combinations) for single-path routing are calculated using MIN-MAX SP and appear in the table below. Also shown are the percentage changes to total of flow between customer-to-customer pairs (C2.0) and flow through the network (*ρ*) after a particular router attack (Y_k =1).

Recall from Chapter II that after attack has occurred, the amount of flow through the network (*ρ*) adjusts to accommodate the “new” capacity and demand constraints.

Table 3. Top 5 Single-Node Attacks Under Single-Path Routing

	Router	Total Flow		ρ	
1	INDIANAPOLIS	431,804	-32%	0.000026	-26%
2	CHICAGO	438,168	-31%	0.000030	-14%
3	ATLANTA	458,729	-27%	0.000028	-20%
4	KANSAS CITY	512,749	-19%	0.000029	-17%
5	WASHINGTON DC	516,005	-18%	0.000051	31%

The top five optimal router attacks to multiple-path routing are obtained by solving MAX MP DUAL. The values from equation (C5.0), the minimized maximum flow, are shown in the next table.

Table 4. Minimized Total Flow

	Router	Minimized Total Flow
1	WASHINGTON DC	98,442
2	NEW YORK	142,047
3	CHICAGO	300,106
4	SEATTLE	333,428
5	INDIANAPOLIS	391,222

We compute attacks 2 through 5 by making the previous router(s) that were attacked invulnerable (i.e., for example, $v_{WashingtonDC,WashingtonDC'}$=0 allows use to determine the second best router attack plan).

After the flow through the network adjusts to accommodate the new capacity and demand constraints after a particular router attack ($Y_{k,k'}$=1), we use R-MAX MP to compute total of flow between customer-to-customer pairs (C6.0) and flow through the network (ρ). Those results are shown in the next table.

Table 5. Top 5 Single-Node Attacks Under Multiple-Path Routing

	Router	Total Flow		ρ	
1	INDIANAPOLIS	431,804	-42%	0.000026	-37%
2	CHICAGO	438,168	-41%	0.000030	-27%
3	ATLANTA	458,729	-38%	0.000028	-32%
4	WASHINGTON DC	516,005	-30%	0.000051	19%
5	NEW YORK	588,184	-20%	0.000055	25%

After flow through the network is adjusted, Indianapolis becomes the worst. Notice, in both models, the changes in total of flow differ from the from the changes in flow through the network, as measured by ρ.

Attacks to Indianapolis, Chicago, and Atlanta are the most devastating. The loss of those routers reduces (but not eliminates, see figure 13) the network's "path diversity" such that the R-MAX MP now only uses single paths when routing traffic.

Notice in the tables above that after a loss of the Washington D.C. router in MIN-MAX SP, and a loss of the Washington D.C. and New York routers in R-MAX MP, flow through the remaining network actually increases. This is again an example of Braess's paradox (discussed in Chapter II). By removing the large demand associated with Washington D.C. customers, $D_{Washington\,DC}$=0 instead of 33,217 Mbps, and it becomes possible to raise flow in the network from .000035(single-path) and .000041(multiple-path) to .000051.

The next table that shows the benefits of R-MAX MP over MIN-MAX SP in terms of total flow in the presence of a router attack.

Table 6. Single-Path vs. Multiple-Path Flow Following a Router Attack

	Single-Path	Multiple-Path	Multiple-Path Increase
Pre-Attack Flows:	630,941	738,442	15%
Post-Attack Flows			
Router Loss:			
ATLANTA	458,729	458,729	0%
CHICAGO	438,168	438,168	0%
DENVER	516,354	642,818	20%
HOUSTON	606,461	619,689	2%
INDIANAPOLIS	431,804	431,804	0%
KANSAS CITY	512,749	592,952	14%
LOS ANGELES	629,785	679,827	7%
NEW YORK	536,860	588,184	9%
SEATTLE	536,483	649,799	17%
SUNNYVALE	583,707	691,886	16%
WASHINGTON DC	516,005	516,005	0%

From the previous example, the next two figures show how data flowing to New York and Sunnyvale is re-routed following the Indianapolis attack.

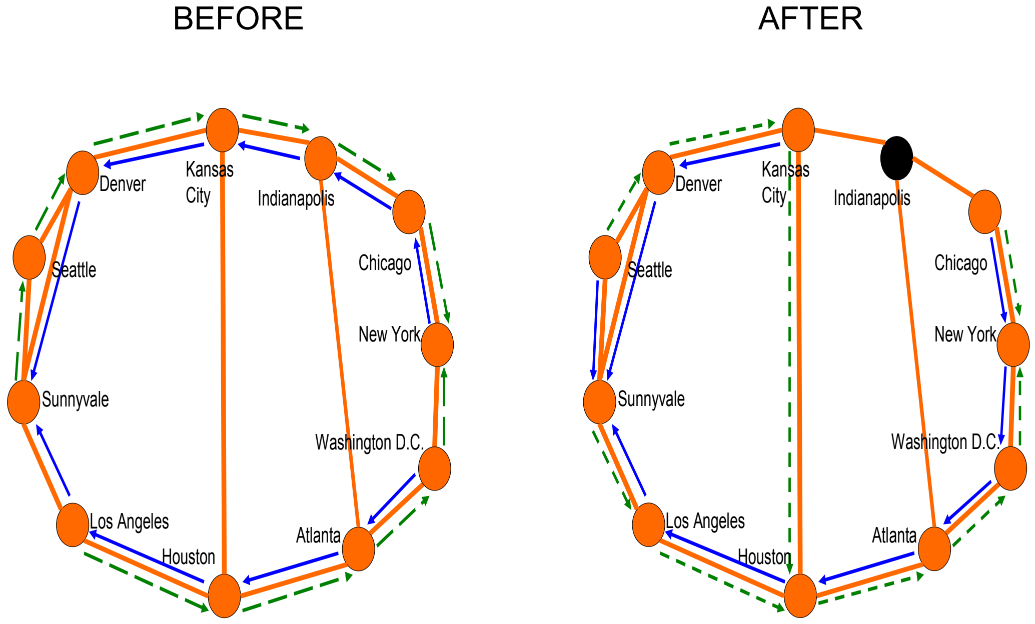
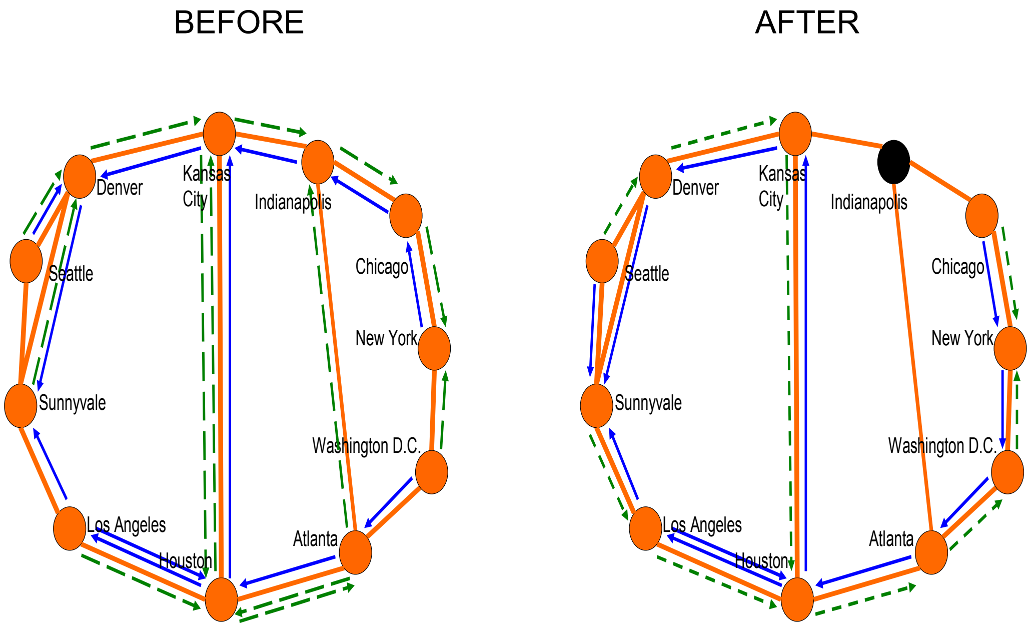


Figure 6. Single-Path Flow to New York & Sunnyvale after Indianapolis Attack


BEFORE

AFTER

Figure 7. Multiple-Path Flow to New York & Sunnyvale after Indianapolis Attack

We observe that R-MAX MP still uses multiple routes when sending traffic to Sunnyvale. However the total flow calculation (C2.0=C6.0=431,804 Mbps) in both models remains the same.

The total amount of traffic traveling to New York and Sunnyvale following the Indianapolis attack is also the same in both models, shown in the table below. By comparing the values to values in Table 2, we observe 27% flow decease in MIN-MAX SP, and a 37% decrease in R-MAX MP.

Table 7. Traffic Flow to New York & Sunnyvale Routers after Indianapolis Attack

		Single-Path & Multiple-Path	
		To	
		NEW YORK	SUNNYVALE
From	ATLANTA	5,398	1,666
	CHICAGO	11,328	3,497
	DENVER	1,593	492
	HOUSTON	1,716	530
	INDIANAPOLIS	------	------
	KANSAS CITY	1,226	379
	LOS ANGELES	9,978	3,080
	NEW YORK	------	7,486
	SEATTLE	15,906	4,910
	SUNNYVALE	7,486	------
	WASHINGTON DC	26,215	8,092

The Washington D.C. transshipment router is the bottleneck in both models.

3. Multiple Node Attacks

Here we extend the previous analysis to the case where the number of router attacks is greater than one (*attacks*>1).

According to MAX MP DUAL, any node attack that splits that network into more than one piece produces an objective value of zero (i.e., (C5.0)= $\sum u_{i,j}\beta_{i,j}$ =0). Thus, a solution formulation is uninformative because we are unable to observe flow through the remaining network (*ρ*). As a result, we compute attacks to the multiple-path network using the R-MAX MP, and we determine which the optimal attacks by total enumeration (just as we do for MIN-MAX MP).

An inspection of figure 1 might lead one to suspect that the optimal two-router attack would split the network in half (i.e., Atlanta and Indianapolis, Kansas City and Houston, etc), or that the optimal two attacks would include Indianapolis, the optimal one-router attack. However, the optimal two-router attack (out of $\binom{11}{2}$ combinations) for both models is Chicago and Seattle.

The resulting network from the Chicago and Seattle attack consists of nine routers and ten arcs. Thus, the only routes that exist between the transshipment routers are the single shortest paths.

The optimal three-router attack (out of $\binom{11}{3}$ combinations) for both models is Chicago, Seattle and Los Angeles.

The next table below shows the total amount of traffic routed between customers, (C2.0) and (C6.0), the flow through the network (ρ), and the transshipment router utilizations (MAX SP: $\frac{\sum_{s,t} q_{s,t}^{i,j}}{B_k}$, MAX MP: $\frac{\sum_{s,t} X_{i,j}^{s,t}}{u_{i,j}}$) in the event of one, two, and three attacks.

Table 8. Optimal Router Attacks

	Single-Path	Multiple-Path		Both Models	
number of Attacks	0	0	1	2	3
Total-Flow	630,941	738,442	431,804	404,454	389,298
ρ	0.000035	0.000041	0.000026	0.00004	0.000051
		Router Utilization			
ATLANTA	0.742	1	0.745	0.710	0.590
CHICAGO	0.652	0.940	0.282	0	0
DENVER	0.578	0.887	0.333	0.066	0.295
HOUSTON	0.608	0.963	0.676	0.534	0.354
INDIANAPOLIS	0.528	0.927	0	0.114	0.125
KANSAS CITY	0.595	1	0.344	0.903	0.326
LOS ANGELES	0.439	0.469	0.382	0.470	0
NEW YORK	0.901	0.939	0.765	0.651	0.294
SEATTLE	0.541	0.633	0.382	0	0
SUNNYVALE	0.335	0.395	0.239	0.239	0.254
WASHINGTON DC	1	1	1	1	1

1 router attack: There is a 32% decrease (i.e., from (C1.0)=630,941 to (C2.0)=431,804 Mbps) and 42% decrease in total amount of traffic routed

between customers, and 26% and 37% decrease in flow through the network in the single-path and multiple-path models, respectively.

1 router attack vs. 2 router attack: In both models, the total amount of traffic routed between customers deceases 6%, while flow through the network increases 35% (Braess's paradox).

2 router attack vs. 3 router attack: In both models, the total amount of traffic routed between customers deceases 4%. Here flow through the network increases 22% (again, Braess's paradox).

The Washington D.C. transshipment router is the bottleneck in both models in all three attacks. Thus, the optimal attacks in all cases do not include the bottleneck. Rather, the attacks seem to redirect flow toward the bottleneck. The bottlenecks restrict flow through the network and thus the attacker does not want to eliminate that restriction.

B. FLOW ON CAPACITATED ARCS

In this section of the analysis, we remove the capacity constraint on the transshipment routers ($B_k = \infty$). Now only the fourteen arcs are capacitated. In reality, the connections are single "duplex" connections, meaning that they support traffic flowing in both directions. Here, we treat each connection as a pair of directed arcs (i.e., Atlanta-to-Houston and Houston-to-Atlanta as different connections), each with a speed of 10 gigabits per second (Gbps) ($u_{i,j}$=10,000 Mbps). The arcs connecting customers to their transshipment router remain un-capacitated.

1. Maximum Flow through Abilene

The tables below show the utilization of twenty-eight transshipment node connections for the single-path ($\frac{\sum_{s,t} q_{i,j}^{s,t}}{u_{i,j}}$) and multiple-path ($\frac{\sum_{s,t} X_{i,j}^{s,t}}{u_{i,j}}$) models, as well as the total amount of flow between customer-to-customer pairs, (C2.0) and (C6.0), and flow through the network (*ρ*).

Table 9. Utilization of Abilene Arcs Under Maximum Flows

	Single-Path	Multiple-Path	Multiple-Path Increase
Total-Flow	67,802	76,467	11%
ρ	0.0000038	0.0000042	10%
ATL-HOUSTON	1	1	0%
ATL-INDY	0.210	0.805	74%
ATL-DC	0.992	1	1%
CHICAGO-INDY	0.757	0.972	22%
CHICAGO-NY	0.781	1	22%
DNVR-KC	0.961	1	4%
DNVR-SEATTLE	0.723	1	28%
DNVR-SUNNY	0.199	0.766	74%
HOUSTON-ATL	1	0.972	-3%
HOUSTON-KC	0.384	0.962	60%
HOUSTON-LA	0.595	0.987	40%
INDY-ATL	0.210	0.834	75%
INDY-CHICAGO	0.757	0.972	22%
INDY-KC	0.620	0.827	25%
KC-DNVR	0.961	1	4%
KC-HOUSTON	0.384	0.934	59%
KC-INDY	0.620	0.855	27%
LA-HOUSTON	0.595	0.987	40%
LA-SUNNY	0.301	0.461	35%
NY-CHICAGO	0.781	1	22%
NY-DC	0.814	0.862	6%
SEATTLE-DNVR	0.723	0.924	22%
SEATTLE-SUNNY	0.168	0.081	-52%
SUNNY-DNVR	0.199	0.843	76%
SUNNY-LA	0.301	0.461	35%
SUNNY-SEATTLE	0.168	0.005	-97%
DC-ATL	0.992	1	1%
DC-NY	0.814	0.862	6%

The next table compares flow through the capacitated-arc network to flow through the capacitated-router network.

Table 10. Arc Capacity vs. Router Capacity

		Single-Path	
Capacity	Arcs	Routers	Decrease
ρ	0.0000038	0.000035	89%
		Multiple-Path	
Capacity	Arcs	Routers	Decrease
ρ	0.0000042	0.000041	90%

The difference in network flow between the capacitated-arc and capacitated-router networks is over 89% for both the MAX SP and MAX MP. Thus arcs are the “severe” constraints on these max flow problems.

Flow levels for the arcs for MAX SP are symmetric, as expected.

There is a 11% increase in total amount of flow between customer-to-customer pairs for MAX MP, and a 10% increase in flow through the network again illustrating the limitations of the single-path traffic routing.

There are only three out of twenty-four instances where an arc is utilized more in the MAX SP than it is in MAX MP: Houston-Atlanta, Seattle-Sunnyvale, and Sunnyvale-Seattle.

The bottlenecks in MAX SP are the Atlanta-Houston and Houston-Atlanta arcs.

There are eight bottlenecks in MAX MP. They are the Atlanta-Houston, Chicago-New York, Denver-Kansas City, Denver-Seattle, Kansas City-Denver, New York-Chicago, Washington D.C.-Atlanta arcs.

2. Single Arc Attack

We examine the impact that losing one of the arcs ($Y_{i,j} = 1$) has on the total amount of flow between customer-to-customer pairs in both models. Remember, destroying arc (*i*, *j*) also destroys arc (*j*, *i*) ($Y_{i,j} = Y_{j,i}$).

Table 11. Single-Path vs. Multiple-Path Flow Following an Arc Attack

	Single-Path		Multiple-Path	
Connection Lost	Total Flow	Net Chg	Total Flow	Net Chg
ATL-HOUSTON	41,866	-38%	41,866	-45%
ATL-INDY	69,701	3%	76,142	-0.4%
ATL-DC	38,234	-44%	38,234	-50%
CHICAGO-INDY	38,773	-43%	38,773	-49%
CHICAGO-NY	38,234	-44%	38,234	-50%
DNVR-KC	42,629	-37%	43,577	-43%
DNVR-SEATTLE	67,802	0%	76,104	-0.5%
DNVR-SUNNY	57,610	-15%	76,467	0%
HOUSTON-KC	68,338	1%	76,467	0%
HOUSTON-LA	43,577	-36%	43,577	-43%
INDY-KC	41,866	-38%	41,866	-45%
LA-SUNNY	53,707	-21%	53,707	-30%
NY-DC	43,183	-36%	43,183	-44%
SEATTLE-SUNNY	67,802	0%	76,104	-0.5%

The optimal attack plan for both models is to attack either the arcs between Atlanta and Washington D.C., or between Chicago and New York.

For MIN-MAX SP, eliminating Atlanta-Indianapolis or Houston-Kansas City slightly increases total flow through the network by 3% and 1% respectively (Braess's paradox).

For R-MAX MP, losing either the Atlanta-Indianapolis arcs (decrease < .4%), Denver-Seattle arcs (decrease < .5%), Denver-Sunnyvale arcs, Houston-Kansas City arcs, or Seattle-Sunnyvale arcs (decrease < .5%) do not have a noticeable impact on total flow through the network. Thus, R-MAX MP is more robust to attacks.

C. FLOW ON CAPACITATED NODES AND ARCS

Now we look at Abilene when both the transshipment routers and their twenty-eight arcs are capacitated.

The routers again have a throughput capacity of 320,000 Mbps. Arc speeds of 10 Gbps produce the same results listed in tables 8 and 9 which implies that the arc capacities are the “real” constraints in the network. Among the eleven transshipment routers, Washington D.C. is utilized the most in both the single-path and multiple-path models with a utilization level of .107 and .115 respectively. To determine the number of arcs in parallel (or bandwidth increase) required to saturate a transshipment router, we uniformly increase the arc capacities until one of the routers saturates.

The results are displayed in the next two figures. The connection speeds are in Mbps.

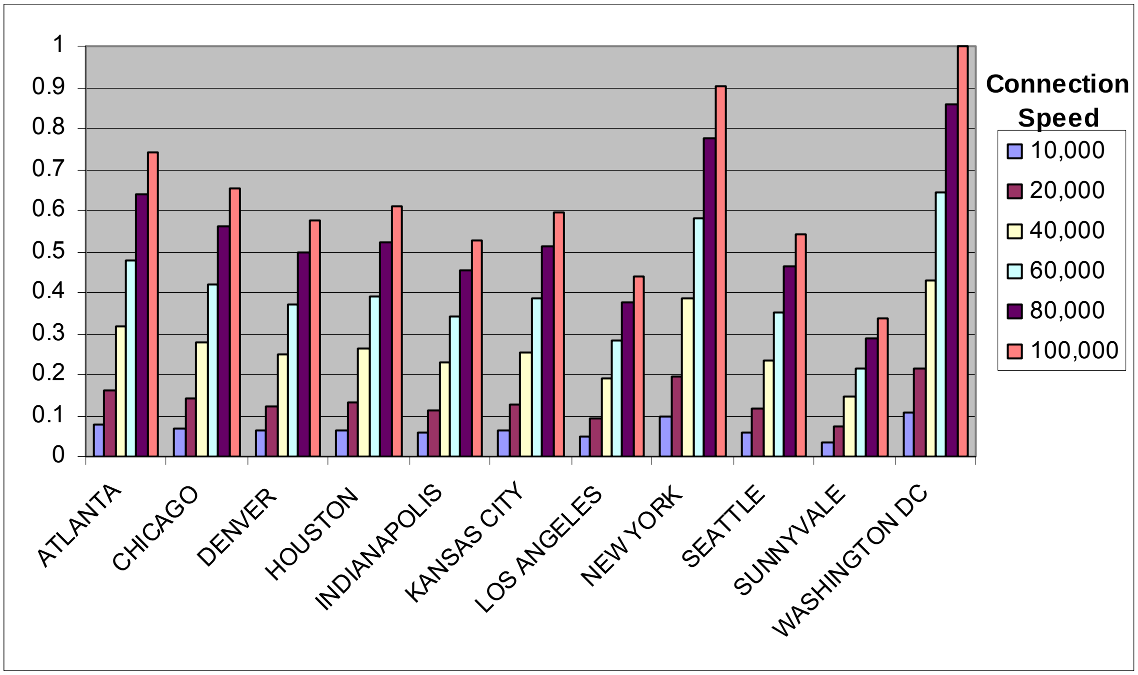


Figure 8. Single-Path Utilization vs. Connection Speed

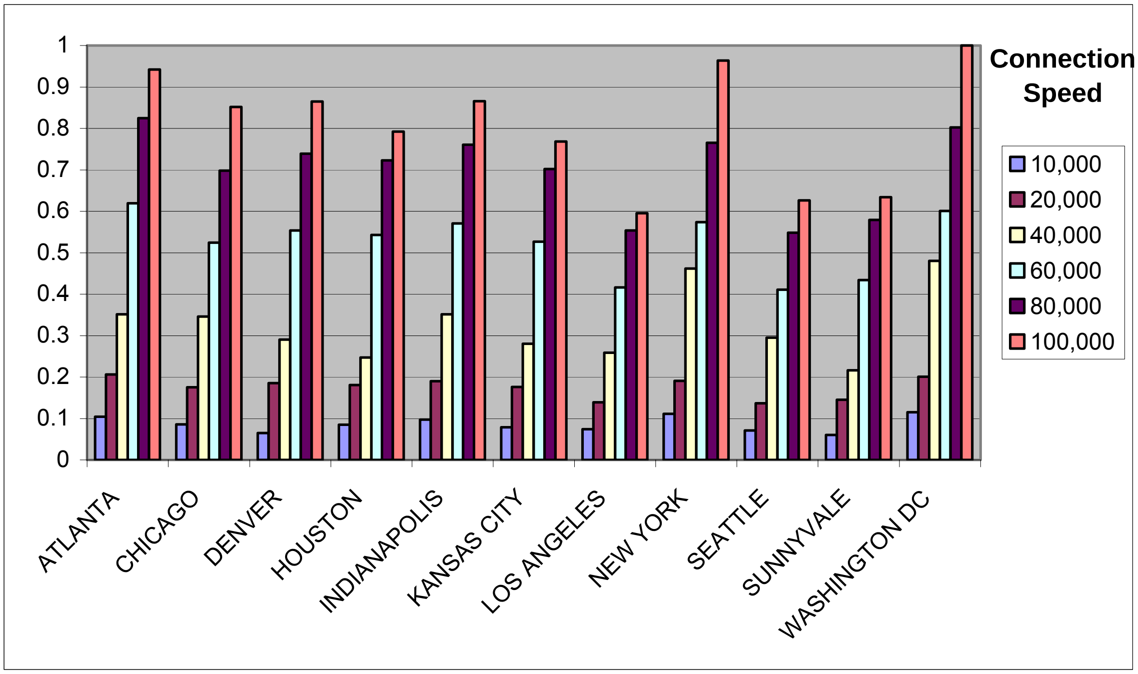


Figure 9. Multiple-Path Utilization vs. Connection Speed

In both models, we can increase arc capacity to 100 Gbps before we saturate a transshipment router. Thus the connection speed required to saturate a router is between 80 and 100 Gbps. Washington D.C. is the router that is saturated in both the single-path and multiple-path models.

D. CHAPTER SUMMARY

We applied the two models representing naive traffic engineering (single-path routing) and best-case traffic engineering (multiple-path routing) to analyze the maximum throughput of Abilene. We performed our analysis with node and/or arc capacities. We found that Abilene is well-provisioned in the sense that it tends to be the arcs, in particular the customer connections, that saturate data flow in the network, a generalization that is consistent with our results.

For both the single-path and multiple-path optimal solutions, the Washington D.C. transshipment router is the bottleneck. Increasing the line speeds of its connections would consequently increase the total amount of flow between customer-to-customer (single-path: $\sum_{s,t} X^{s,t}$, multiple-path: $\sum_{s,t} Z^{s,t}$) and flow through the network (ρ).

Our interdiction analysis shows that the optimal transshipment router attack is to remove Indianapolis. The second worst single router attack is Chicago, which is involved in both the optimal two-router (Chicago, Seattle) and three-router (Chicago, Los Angeles, Seattle) attacks. The optimal single arc attack is either the Atlanta-Washington D.C. or Chicago-New York arc. These results suggest the importance of the Indianapolis and Chicago routers to Abilene. Perhaps this is where redundancy should be built into the network.

We conclude that Abilene is “over provisioned” in terms of its routers and can handle increasing connection speeds (i.e., multiple connections in parallel). Line speeds of 40 Gbps (OC-768) could be implemented without needing to upgrade the routers.

IV. SUMMARY AND CONCLUSIONS

The models and analysis in this thesis are applicable to any ISP network. The general public, businesses, civilian and military organizations rely heavily on these networks. As the reliance grows, so will the need for understanding an ISP's limitations and vulnerability to attacks.

The interdiction models used in this thesis MIN-MAX SP and MAX MP Dual, identify the attack plan that reduces the maximum amount of traffic carried among users in the worst possible way. One could take our analysis a step further by conducting defender-attacker-defender analysis where the *defender* (network operator) first decides which network components to protect, and study how those decisions effect the *attacker's* plan (Brown et al. 2006).

Future work will need to study alternatives to the gravity model because of the difficulties that arise when using it. For any multi-commodity gravity model network, flow through the network goes to zero if a single node cannot meet its demand. So interdicting a multi-commodity gravity model network is simple, just "disconnect" any node from the network. We avoid this in the single-path model by solving for the *ρ* for each node *k* algebraically where B_k >0. Our revised multiple-path formulation (figure 6) allows us to get around that for the multiple-path models. However, as demonstrated in Chapter III, there are often cases where Braess's paradox occurs. It is unsettling that flow through the network could rise after an attack.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A: FLOYD-WARSHALL ALGORITHM

The Floyd-Warshall algorithm computes the shortest path between each pair of nodes in a network (R. Ahuja et al. 1993). The algorithm builds an $n \times n$ matrix of shortest path distances, d(i, j), for each node pair, as well as an $n \times n$ matrix of predecessor nodes, pred(i, j), for each node in a particular path. If no path exists, the distance is reported as ∞, and the predecessor is null. Floyd-Warshall runs in O(n^3)operations.

```
algorithm Floyd-Warshall;
    begin
        for all node pairs $(i, j) \in N \times N$ do
            $d(i, j) := \infty$ and $pred(i, j) := 0$;
        for all nodes $i \in N$ do $d(i, i) := 0$;
        for each arc$(i, j) \in A$ do $d(i, j) := nC$ and $pred(i, j) := i$;
        for each node $k$: 1 to $n$ do
            for each $(i, j) \in N \times N$ do
            if $d(i, j) > d(i, k) + d(k, j)$ then
            begin
                $d(i, j) := d(i, k) + d(k, j)$;
                $pred(i, j) := pred(k, j)$;
            end;
    end;
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B: COMPUTING $R^k_{s,t}$

The following is the algorithm builds a matrix (e(e-1) rows × n columns) of ${r^k}_{s,t}$ values (binary) for use in the single-path formulations. It uses as input the pred(*i*, *j*) matrix obtain from Floyd-Warshall. The algorithm runs in $e(e\text{-}1) \cdot n$ operations.

```
algorithm Compute ${r^k}_{s,t}$ ;
begin
        for all (s, t) $\in$ $E$ x $E$ do
                for all $k \in N$ do
                        $R^k_{s,t}$=0;
end;
begin
        for s $\in$ {1, 2,…, $E$} do
                for t $\in$ {1, 2,…, $E$} do
                        k = $t$;
                        do {
                                ${r^k}_{s,t}$ = 1;
                                k = pred($s$, $k$);
                        } while (k $\neq s$)
end;
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C: COMPUTING $q_{i,j}^{s,t}$

The following is the algorithm builds a matrix (e(e-1) rows x m columns) of $q_{i,j}^{s,t}$ values (binary) for use in the single-path formulations. The algorithm uses as input the pred(*i*, *j*) matrix obtained from Floyd-Warshall. The algorithm runs in $s(s\text{-}1) \cdot m$ operations.

```
algorithm Compute $q_{i,j}^{s,t}$ ;
begin
    for all (s, t) $\in$ $E$ x $E$ do
        for each arc (i, j) $\in$ A do
            $q_{i,j}^{s,t}$ =0;
end;
begin
    k = t;
    do {
        if pred(s, k) = j then;
            if pred(s, j) = i then;
                $q_{i,j}^{s,t}$ = 1;
                k = pred(s, i);
    } while (k $\neq$ s)
end;
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX D: STANDARD LP MAX FLOW

The standard LP maximum flow formulation adds a unbounded return arc, arc (*t*, *s*), to a network and then maximums flow on that arc. The rest of the formulation consist of balance of flow and arc capacity constraints.

Index Use

$i,\ j \in N$	Nodes
$(i, j) \in A$	Directed arc from node *i* to node *j*
s, t	Source and terminal nodes

Data

$u_{i,\ j}$	Upper bound on flow from node *i* to node *j* on arc (*i*, *j*) $\in$ *A* [flow]

Decision Variables

$X_{i,\ j}$	Flow on directed arc (*i*, *j*) $\in$ *A* [flow]

Formulation

$$\max \quad X_{t,s}$$

$$s.t. \sum_{(j,i)\in A} X_{j,i} - \sum_{(i,j)\in A} X_{i,j} = 0 \qquad \forall i \in N$$

$$0 \le X_{i,j} \le u_{i,j} \qquad \forall (i, j) \in A$$

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX E: REACHABILITY FORMULATION / ALGORITHM

We determine if source node *s* is able to reach terminal node *t* after an attack has occurred for all *s*-*t* pairs in the network. Since we solve the multiple-path formulations in GAMS, we also use GAMS to determine node reachabilty by solving another LP formulation (as opposed to a standard reachability algorithm which GAMS would execute very slowly).

Index Use

$i,\ j,\ k \in N$	Nodes
$(i, j) \in A$	Directed arc from node *i* to node *j*
$s, t \in E \subseteq N$	Source and terminal nodes in the set of "edge" nodes *E*

Data

$Y^*_{i,j}$	Binary indicator whether the attacker destroyed of arc (*i*, *j*) $\in$ *A*

$$Y^*_{i,j} = \begin{cases} 1 \text{ if } arc\,(i,\ j)\,\text{was destroyed} \\ 0 \text{ otherwise} \end{cases}$$

Decision Variable

$W_{s,t}$	Flow from node *s* to node *t* [Flow]
$Q^s_{i,j}$	Flow from node *s* on arc (*i*, *j*) $\in$ *A* [Flow]

Formulation

$$\max \sum_{s,t \in E} W_{s,t}$$

$$s.t. \sum_{(i,j)\in A|Y^*_{i,j}=0} Q^s_{i,j} - \sum_{(i,j)\in A|Y^*_{i,j}=0} Q^s_{j,i} = \begin{cases} \sum_j W_{i,j} \text{ if } s=i \\ 0 \\ -W_{s,j} \text{ if } s \neq i \end{cases} \qquad \forall s \in E, i \in N$$

Discussion

By maximizing $W_{s,t}$ for all *s* and *t,* we simultaneously determine whether we can send flow from *s* to *t* along the "surviving" arcs in the network ($Q^s_{i,j}$). The dual of the multiple-path maximum flow model (figure 5) determines the $Y^*_{i,j}$ values. $Y^*_{i,j}$=1 implies arc (i, j) is attacked and thus has zero capacity for flow, and $Y^*_{i,j}$=0 implies arc (i, j) survived the attack.

Once we have established which nodes *t* are reachable from which nodes s (i.e., if a path from *s* to *t* exist), we are able to compute the $R^{s,t}$ values used in the revised multiple-path max flow model (figure 6) with a simple algorithm.

```
algorithm Compute $R^{s,t}$ ;
begin
    for all (s, t) $\in$ E do
        if $W^*_{s,t}$>0 then;
            $R^{s,t}$=1;
        else $R^{s,t}$=0;
end;
```

LIST OF REFERENCES

[1] J. Rexford (2006). “Route Optimization in IP Networks.” Handbook of optimization in telecommunications. Springer.

[2] D. Alderson, H. Chang, M. Roughan, S. Uhlig, and W. Willinger (2006). “The Many Facets of Internet Topology and Traffic.” Networks and Heterogeneous Media, American Institute of Mathematical Sciences, Volume 1, Number 4, pp. 569-600.

[3] D. Alderson, L. Li, W. Willinger, and J. Doyle (2005). “Understanding Internet Topology: Principle, Models, and Validation.” IEEE/ACM Transactions on Networking, Volume 13, Number 6, pp. 1205-1218.

[4] J. Doyle, D. Alderson, L. Li, S. Low, M. Roughan, S. Shalunov, R. Tanaka, and W. Willinger (2005). “The “robust yet fragile” nature of the Internet.” The National Academy of Sciences of the USA, Volume 102, Number 41, pp. 14497-14502.

[5] M. Henningsson, K. Holmberg, and D. Yuan (2006). “Ring Network Design.” Handbook of optimization in telecommunications. Springer.

[6] M. Grötschel, C.L. Monma, and M. Stoer (1995). “Design of Survivable Networks.” Handbooks in operations research and management science, Network Models, Chapter 6, Number 7. Elsevier.

[7] H. Kerivin and A. R. Mahjoub (2005). “Design of Survivable Networks: A Survey.” Networks, Volume 46, Issue 1, pp. 1-21.

[8] R. Albert, H. Jeong, and A. L. Barabási (2000). “Attack and Error Tolerance of Complex Networks.” Nature, Volume 406, pp. 378-382.

[9] R. Cohen, K. Erez, D. ben-Avraham, and S. Havlin (2000). “Resilience of the Internet to Random Breakdowns.” Physical Review Letters, 85(21), pp. 4626-4628.

[10] R. Cohen, K. Erez, D. ben-Avraham, and S. Havlin (2001). “Breakdown of the Internet Under Intentional Attack.” Physical Review Letters, 86(16), pp. 3682-3685.

[11] B. Bollobás and O. Riordan (2003). “Robustness and Vulnerability of Scale-Free Random Graphs.” Internet Mathematics, Volume 1, Number 1, pp. 1-35.

[12] P. Crucitti, V. Latora, M. Marchiori, and A. Rapisarda (2004). “Efficiency of Scale-Free Networks: Error and Attack Tolerance.” Physical Review Letters, Edition 69, 045104(R).

[13] D. Alderson (2008). "Catching the 'Network Science' Bug: Insight and Opportunity for the Operations Researcher." Operations Research. In press.

[14] G. Brown, M. Carlyle, J. Salmeron, K. Wood (2006). “ Defending Critical Infrastructure.” Interfaces, Volume 36, Number 6, pp. 530-544.

[15] R. Rardin (1998). “Optimization in Operations Research.” Pearson Education Inc. Delhi India, pp. 627-628.

[16] M. Florian and D. Hearn (1995). “Network Equilibrium Models and Algorithms.” Handbook in operations research and management science, Network Routing, Elsevier, Chapter 6, Number 8, pp. 485-550.

[17] R. Ahuja, T. Magnanti, and J. Orlin (1993). “Network Flows.” Prentice-Hall Inc., Upper Saddle River, NJ, pp. 147-148.

INITIAL DISTRIBUTION LIST

1. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

2. David Alderson
   Naval Postgraduate School
   Monterey, California

3. Gerald Brown
   Naval Postgraduate School
   Monterey, California

4. Matthew Carlyle
   Naval Postgraduate School
   Monterey, California